Apache directive	Nginx equivalent
`ServerTokens`: Apache allows you to configure the information transmitted in the request headers regarding the server OS and the software name and versions.	`server_tokens`: In Nginx, you may enable or disable the transmission of server information by using the `server_tokens` directive from the main HTTP module.
`ServerRoot`: This lets you define the root folder of the server, which contains the configuration and the logs folder.	The `--prefix` build-time option: With Nginx, this option is defined at compile time with the `--prefix` switch of the configure script, or at execution time with the `-p` command line option.
`PidFile`: This defines the path of the application PID file.	`pid`: The exact equivalent directive is `pid`.
`TimeOut`: This directive defines three elements: • The maximum execution time of a GET request. • The maximum allowed delay between two TCP packets in POST and PUT requests. • The maximum allowed delay between two TCP ACK packets.	Multiple directives: There are multiple directives allowing a similar behavior: `send_timeout`: This defines the maximum allowed delay between two read operations by the client. `client_body_timeout`: This defines the timeout for reading the client request body. `client_header_timeout`: This defines the timeout for reading the client request headers.
`KeepAlive`, `MaxKeepAliveRequests`, `KeepAliveTimeout`: These three directives control the keep-alive behavior of Apache.	`keepalive_timeout`, `keepalive_requests`: These two directives are the direct equivalents of the Apache ones, except that if you want to completely disable the keepalives, set `keepalive_timeout` or `keepalive_requests` to `0`.
`Listen`: This defines the interface and port on which Apache will listen for connections.	`listen`: In Nginx, this directive is only defined at the virtual host level (`server` block).
`LoadModule`: With this directive, Apache offers the possibility to load the modules dynamically.	`--with_****_module`: Nginx cannot load modules dynamically; these need to be included at compile time. Once incorporated in Nginx, they cannot be disabled.
`Include`: A file inclusion directive that supports wildcards.	`include`: The `include` directive of Nginx is identical.
`User`, `Group`: Allows you to define the user and group under which the daemon will be running.	`user`: The `user` directive of Nginx lets you specify both the user and the group.

Apache directive	Nginx equivalent
`ServerAdmin`, `ServerSignature`: This lets you specify the e-mail address of the server administrator as well as a signature message to be displayed on error and diagnostic pages.	No equivalent As of version 1.8, there is no equivalent in Nginx. Error pages do not show the e-mail address of the server administrator or other information. Use the `error_page` directive to customize your site's error pages.
`UseCanonicalName`: This defines how Apache constructs self-referential URLs.	No direct equivalent Although there is no direct equivalent for this Apache directive, the construction of self-referential URLs can be defined via module-specific settings (proxy, FastCGI, and so on).
`DocumentRoot`: This defines the root folder from which Apache will serve files. The directive can be used at the server and virtual host levels.	`root`: The `root` directive can be inserted to define the document root at all levels: `http`, `server`, `location`, and `if` blocks.
`DirectoryIndex`, `IndexOptions`, `IndexIgnore`: Defines the directory index and file listing options.	`index`, `autoindex`, `random_index`, `fancyindex` (third party): Nginx also offers a good variety of options for managing indexes.
`AccessFileName`: Defines the filename of `.htaccess` files that are included dynamically on page execution.	No equivalent Nginx, as of version 1.8, has no such feature as `.htaccess` files. Read further sections for more information.
`TypesConfig`, `DefaultType`: Defines the MIME type options.	`types`, `default_type`: Equivalent directives exist in Nginx, although with a different syntax.
`HostNameLookups`: This allows looking up of hostnames for client IP addresses for logging or access control purposes.	No equivalent As of Nginx 1.8, there is no equivalent functionality.
`ErrorLog`, `LogLevel`, `LogFormat`, `CustomLog`: Logging activation and format settings.	`access_log`, `log_format`: Nginx also allows a large variety of options, but they are combined in fewer directives.
`Alias`, `AliasMatch`, `ScriptAlias`: Directory aliasing options.	`alias`: The `alias` equivalent directive is offered by Nginx, but nothing for the other two.